

Quantization Effects in Viterbi Decoding Rate $1/n$ Convolutional Codes

I. M. Onyszchuk, K.-M. Cheung, and O. Collins
Communications Systems Research Section

A Viterbi decoder's performance loss due to quantizing data from the additive white Gaussian noise (AWGN) channel is studied. An optimal quantization scheme and branch metric calculation method are presented. The uniformly quantized channel capacity $C_u(q)$ is used to determine the smallest number of quantization bits q that does not cause a significant loss. The quantizer stepsize which maximizes $C_u(q)$ almost minimizes the decoder bit error rate (BER). However, a slightly larger stepsize is better, like the value that minimizes the Bhattacharyya bound. The range and renormalization of state metrics are analyzed, in particular for $K=15$ decoders such as the Big Viterbi Decoder (BVD) for the Galileo mission. These results are required to design reduced hardware complexity Viterbi decoders with a negligible quantization loss.

I. Introduction

Theoretically, Viterbi decoding is a maximum-likelihood decoding algorithm for convolutional codes. In practice, the main performance loss results from quantizing input data with q bits. The decoder's hardware complexity and speed depend strongly upon q and the state metric register length ℓ . Therefore, these parameters must be chosen as the smallest values that do not cause a significant bit signal-to-noise ratio (E_b/N_0) loss. A constraint length $K=15$ decoder performs double the computation of a $K=14$ decoder, but requires about 0.1 dB less E_b/N_0 for a bit error rate (BER) of 0.005. Since part of the decoder's hardware complexity increases only linearly with q , even a 0.01-dB quantization loss is large. However, given

that one must construct a fully parallel $K=15$ (or $K=7$) decoder, a slightly larger loss might be acceptable or required by hardware and speed constraints.

The uniformly quantized, additive white Gaussian noise (AWGN) channel capacity $C_u(q)$ is used to estimate the quantizer stepsize Δ and smallest q that result in a negligible loss. For each q , almost minimum BER occurs when Δ maximizes $C_u(q)$ or minimizes the Bhattacharyya bound γ . New methods are presented to minimize the state metric register length ℓ in bits. These estimates are verified by simulations of three codes: the constraint length $K=7$, rate $R=1/2$, NASA standard code; the new experimental $K=15$, $R=1/4$, Galileo code [1]; and the $K=15$, $R=1/6$, "2-dB" code [2].

These results are used to determine the best design parameters q , ℓ , and Δ for $K=15$, rate 1/4 and rate 1/6 decoders. Using 6 quantization bits with 10-bit state metric registers would substantially reduce the Big Viterbi Decoder (BVD) [4] hardware complexity and allow the system clock frequency to decrease by a factor of 0.56 as compared to the current design, which has $q = 8$ and $\ell = 16$. Using $q=5$ would cause an E_b/N_0 loss of 0.02 dB at a BER of 0.005 for the Galileo code or “2-dB” code, but there is no measurable E_b/N_0 loss when $q = 6$. Since the same losses occurred for 8-bit symbol error rates (SER), these results apply when an outer block code is concatenated with the convolutional code.

II. Branch Metrics

When an encoded 0 or 1 is mapped to +1 or -1, respectively, and then transmitted, the receiver’s demodulator output is a conditionally Gaussian random variable y with mean $+m$ or $-m$ and the same variance $\sigma^2 = m^2/(2RE_b/N_0)$ as the zero-mean AWGN channel noise. (This holds for binary phase shift-keyed [BPSK] signaling with ideal coherent detection.)

For the AWGN channel, a Viterbi decoder finds the trellis path with minimum Euclidean distance (or equivalently, minimum negative inner product) to the received sequence. Thus, each trellis branch metric is the inner product of the length n branch label (with 0 and 1 replaced with +1 and -1) and the negative of a received vector $[y_1, y_2, \dots, y_n]$. Hence, the decoder adds $-y_i$ or $+y_i$ (equivalently $(-y_i + |y_i|)/2$ or $(y_i + |y_i|)/2$ when σ is fixed, because incrementing or multiplying all branch metrics by a constant does not change the decoder’s output) to the metrics of those branches with a +1 or -1 in position i . Therefore, the decoder may add $|y_i|$ to the metrics of branches having different signs in position i than that of y_i , and zero otherwise. This sign-magnitude method is used throughout this article because it halves the branch and state metric maximum ranges, as compared to using standard integer metrics [3,4]. For example, using this method in the Scarce-State $K = 7$, rate 1/2 decoder [5] would substantially decrease the chip circuitry.

III. Quantization

When zeros and ones are equally likely in the encoder input data,

$$\Pr(|y| = x) = \frac{1}{2} [\Pr(y = x | +1) + \Pr(y = -x | +1)]$$

$$\begin{aligned} & + \frac{1}{2} [\Pr(y = x | -1) + \Pr(y = -x | -1)] \\ & = \Pr(y = x | +1) + \Pr(y = x | -1) \\ & = \frac{1}{\sqrt{2\pi}\sigma} \left[e^{-(x-m)^2/2\sigma^2} + e^{-(x+m)^2/2\sigma^2} \right] \end{aligned}$$

In this article, $m = 0.84$ volts. The probability distribution function of $|y|$ (Fig. 1) suggests that more quantization levels are required for the $K = 15$ codes operating near 0 dB (high noise variance) than the NASA code at $E_b/N_0 = 2.25$ dB.

Let the random variable J be the quantized value of y and for $-2^{q-1} - 2 \leq j \leq 2^{q-1} - 2$ define

$$\begin{aligned} p_j &= \Pr(J = j | +1) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \int_{(j-0.5)\Delta}^{(j+0.5)\Delta} e^{-(y-m)^2/2\sigma^2} dy \end{aligned}$$

For $j = \pm(2^{q-1} - 1)$, p_j is the above integral with limits $(j - 0.5)\Delta$ and $+\infty$, or $-\infty$ and $(j + 0.5)\Delta$.

Since $|J_1|, \dots, |J_n|$ are summed to form branch metrics, the absolute error $|J_i - y_i|$ in quantizing y_i is also the contribution to the branch metric error incurred. A decoder using signed integers to represent J_i could conceptually use $0, \pm\Delta, \pm2\Delta, \dots, \pm(2^{q-1} - 1)\Delta$ for any real number Δ , because multiplying all metrics by Δ has no effect. Therefore, the quantizer thresholds should be uniformly spaced Δ volts apart at $\pm\Delta/2, \pm3\Delta/2, \dots, \pm(2^{q-1})\Delta/2$, because this minimizes the metric error defined above (and also any positive function of $J_i - y_i$). Thus, only uniform quantization schemes, characterized by q and Δ , are considered herein. (Several simulations of the NASA code using 3-bit integer branch metrics and nonuniform quantization schemes never produced lower BERs than using the best Δ).

For $q = 3$, J_i is normally one of 7 values from -3 to +3, so quantizer levels +4 and -4 are appended (Fig. 2) to decrease the BER near that for 8 levels and standard integer metrics. Thus, the maximum magnitude of J , $2^{q-1} - 1$, will be replaced by 4 instead of 3 for $q = 3$ throughout this article. In rate 1/2 decoders, a branch metric of 8 is decreased to 7 so that $q = 3$ bits still represent all possible values. Since $\Pr(|J_i| = j) = p_j + p_{-j}$, this event occurs with probability $(p_{+4} + p_{-4})^2$, which is only 0.11 for the NASA code at $E_b/N_0 = 2.25$ dB.

IV. Quantizer Stepsize

Ideally, the uniform quantizer stepsize Δ should minimize BER and SER over the decoder's operating range of channel noise levels. In practice, a Δ which almost minimizes the BER for the lowest expected E_b/N_0 will also nearly minimize both the BER and SER when E_b/N_0 increases by up to 1 dB. Simulations (described later) indicate that the Δ that maximizes channel capacity is near optimum.

Since the binary-input quantized AWGN channel is symmetric, capacity is achieved with equiprobable inputs:

$$C_u(q) = \sum_{j=-2^{q-1}+1}^{2^{q-1}-1} p(j|+1) \log_2 \left[\frac{2p(j|+1)}{p(j|+1) + p(j|-1)} \right]$$

$$= 1 - \sum_{j=-2^{q-1}+1}^{2^{q-1}-1} p_j \log_2 \left(1 + \frac{p_{-j}}{p_j} \right)$$

bits per channel use. Figure 3 shows how rapidly the maximum possible uniformly quantized AWGN channel capacity $C_u(q)$ approaches its limit for several noise variances; $C_u(3)$ is based upon the 9-level quantizer in Fig. 2 instead of using 8 or 7 levels. A $q = 4$ or $q = 5$ quantizer has 15 or 31 levels, respectively. The data points in Fig. 2 indicate $C_u(q)$ for integer values of q . The lines between data points are channel capacities when uniform quantizers have intermediate numbers of levels, such as 24.

The curves in Fig. 3 show that there is negligible capacity gain for $q > 6$, and in fact $C_u(5)/C_u(\infty) \geq 0.9975$ suggests that there will be a very small loss for $q = 5$. Figure 4 shows how the performance of the NASA code at $E_b/N_0 = 2.25$ dB varies with q and Δ . Observe that the minimum BER for $q = 5, 4$, and 3 increases roughly in proportion with the decrease in capacity. Also, for $q = 5$, there is a negligible loss and the BER increases extremely slowly for Δ greater than the optimum. Therefore, for $q \geq 4$, it is important to choose Δ larger instead of smaller than the best value. The labels C and γ in Fig. 4 indicate the stepsizes that respectively maximize $C_u(q)$ and minimize the Battacharyya bound parameter

$$\gamma = \sum_{j=-2^{q-1}+1}^{2^{q-1}-1} \sqrt{p_j p_{-j}}$$

which is a measure of the channel noise level: near 0 for high E_b/N_0 and approaching 1 for very noisy channels.

The Δ that minimizes γ is the safest choice because it is slightly larger than the stepsize which minimizes BER. Also, minimizing γ yields the lowest BER for $q = 3$ with 9 quantizer levels (Fig. 4). Finally, the corresponding 8-bit SER curves are not shown because they have the same relative shape and spacing as the BER curves in Fig. 4. Many sets of software simulations were run for the NASA code and the Galileo code. The values of q were 3, 4, 5, or 6 and E_b/N_0 ranged from 0 dB to 3.5 dB.

In all simulations, the Δ s which maximize $C_u(q)$ or minimize γ were, respectively, slightly smaller or larger than the Δ that minimized BER. For $q = 3$ or 4 , the Δ s which minimize the quantizer mean-square error or absolute error were too large.

The simulations in Fig. 5 for the $K = 15$ codes show that using $q = 5$ or 4 costs 0.02 dB or 0.05 dB at the BER of 0.005 required for images. These E_b/N_0 quantization losses are the same when the Viterbi decoder output becomes the input to an outer block decoder, because the 8-bit symbol error rate curves are spaced the same distance apart as the BER curves. In all simulations, the uniform spacing Δ was chosen to minimize γ .

V. State Metric Renormalization

For each received n -vector and encoder state, a Viterbi decoder finds the trellis path with least total branch metrics into the state. Since the state metrics are stored in ℓ -bit registers, occasionally they must all be decreased to avoid overflow. This renormalization can be accomplished by zeroing every register's most significant bit (msb), which is equivalent to subtracting $2^{\ell-1}$ from every metric if all registers have msb = 1. However, detecting when all 2^{K-1} metrics simultaneously have msb = 1 is impractical for a $K = 15$ decoder such as the BVD.

At each trellis level, let the random variable M be the difference between the maximum and minimum state metrics. If any state metric is $\geq 2^{\ell-1} + 2^{\ell-2}$, (its two most significant bits are 1) and $M < 2^{\ell-2}$, then all metrics are $\geq 2^{\ell-1}$, so every msb = 1. In the BVD, $\ell = 16$ was chosen to guarantee that $2^{\ell-2} > M$, and so a single state metric is monitored and renormalization occurs when the two most significant bits are 1. The following improved method should be used when ℓ is reduced so that $M \geq 2^{\ell-2}$. Let W be the maximum of the metrics of the all-zeros state, the all-ones state, and the state with a one input followed by $K - 2$ zeros. Since most state metrics differ from one of these three metrics by only a few $|J_i|$ contributions, W is close to the largest state metric

(Galileo code simulations verified this). Therefore, renormalization could occur when W exceeds a threshold such as $2^{\ell-1} + 2^{\ell-2} + 2^{\ell-3}$. If more metrics are monitored, the threshold can be set closer to $2^\ell - 1$ because W will be closer to the largest state metric.

Definition. Let D be the maximum, over all nonzero states s , of the least-weight trellis path from the all-zeros state into state s .

Lemma. $M \leq D(2^{q-1} - 1)$

Proof. Let b and w be the states with lowest and highest metrics. Since a convolutional code is linear, there exist two trellis paths from some state c , one into state b and the other into state w , whose branch labels differ in D or fewer positions. Since the maximum contribution to a branch metric by one J_i is $2^{q-1} - 1$, the state metric of w is at most the state metric of b plus $D(2^{q-1} - 1)$.

Corollary. In the absence of noise,

$$M = M_0 = D \cdot j_m$$

where $j_m = \lfloor 0.5 + m/\Delta \rfloor$ is the quantizer output when $+m$ volts is input.

For nonsystematic codes, D is near d_{free} and usually much less than $n(K-1)$, the maximum possible. The NASA code has $d_{\text{free}} = 10$, $D = 8$, and $n(K-1) = 12$. Since $D = 33$ for the Galileo code, $M_0 = 132$ for $q = 5$, $\Delta = 0.20$, and $m = 0.84$. Since $D = 50$ for the rate 1/6 "2-dB" code, $M_0 = 200$. Simulations for the Galileo and NASA codes show that M_0 is an upper bound on the mean of M when the channel is noisy and $2M_0$ is always greater than M .

As in the $q = 3$ case where levels $+4$ and -4 were adjoined, a rule for limiting branch metrics may be derived by computing their probabilities. Define

$$m(x) = p_0 + \sum_{j=1}^{2^{q-1}-1} (p_j + p_{-j})x^j$$

Then $\Pr(|J_i| = j) = \{m(x)\}_j$, the coefficient of x^j in $m(x)$. Since the largest possible branch metric is the sum of n independent values $|J_i|$, \dots , $|J_n|$, it equals t with probability $\{[m(x)]^n\}_t$. Thus M could be reduced by limiting branch metrics.

$$\text{Claim. } \Pr(M \geq t) \leq \sum_{i=t}^{D(2^{q-1}-1)} \{[m(x)]^D\}_i$$

where the subscript i denotes the coefficient of x^i in the polynomial within the braces.

Proof. Let b and w be the states with lowest and highest metric. An upper bound on $\Pr(M=t)$ is obtained by considering the worst possible case: the survivor path for state w differs from that for state b in exactly D positions, and in these positions, the survivor path branch labels of w have a different sign than the received J_i . Then $\Pr(M=t)$ is the coefficient of x^t in $[m(x)]^D$.

To achieve a particular (very low) probability, t must be unrealistically large since the above bound is not very tight. This is fine, because t could be chosen as the least power of 2 such that $\Pr(M \geq t) \leq 10^{-5}$. Then setting $\ell = 1 + \log_2 t$ results in no loss of performance.

The current BVD design has $q = 8$ and $\ell = 16$ to accommodate $M \leq n(K-1)(2^{q-1}-1)$ and two extra bits for renormalization. This results in full maximum-likelihood decoder performance. However, using $q=6$, $\Delta = 0.14$, and $\ell = 10$ for the BVD operating at 0 to 1 dB E_b/N_0 would not increase the BER or SER detectably, but would reduce the decoder hardware. Furthermore, the system clock frequency and thus timing constraints would be reduced by a factor of 10/18.

When $\ell < 1 + \log_2[D(2^{q-1}-1)]$, then occasionally a state metric may overflow, whereupon it is immediately decreased by 2^ℓ , instead of $2^{\ell-1}$ at the next renormalization. Protecting against overflow is important because a state with a high metric might suddenly become one of the best states, causing the decoder to make wrong decisions. This can be avoided by setting state metrics that overflow equal to all ones ($2^\ell - 1$). Then states with very high metrics remain this way even after renormalization so they do not affect the decoder's output. An underflow is the event that occurs at renormalization when a state metric has $\text{msb} = 0$, in which case the metric is effectively increased by $2^{\ell-1}$. Rarely, underflows may occur because it is infeasible to continuously check all 2^{K-1} state metrics to find the least value. In conclusion, overflows can be prevented by extra hardware, but underflows will occasionally happen. In practice, always examining several state metrics gives a good approximation of the current metric size and range M . Hence, renormalization can take place so that overflows and underflows occur with very low probability.

Myth. When state metrics overflow or underflow, the decoder fails completely.

One million decoded bit simulations for $q = 5$ and 4 with short state metric registers having $\ell = 9$ and 8 bits,

respectively, yielded the same results as in Fig. 5, because the odd underflow or overflow that occurred did not significantly affect the output. This follows from the Viterbi decoder's robustness and tolerance of occasional state metric disruptions. Further shortening of the state metric regis-

ters to 8 and 7 bits resulted in a graceful BER increase, as though q was being decreased. This behavior is expected because the overall trellis path metric resolution is the decoder parameter, affected by input quantization, which influences decisions.

References

- [1] J. Statman, G. Zimmerman, F. Pollara, and O. Collins, "A Long Constraint Length VLSI Viterbi Decoder for the DSN," *TDA Progress Report 42-95*, vol. July–September 1988, Jet Propulsion Laboratory, Pasadena, California, pp. 134–142, November 15, 1988.
- [2] J. H. Yuen and Q. D. Vo, "In Search of a 2-dB Coding Gain," *TDA Progress Report 42-83*, vol. July–September 1985, Jet Propulsion Laboratory, Pasadena, California, pp. 26–33, November 15, 1985.
- [3] J. A. Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communication," *IEEE Trans. Commun. Tech.*, vol. COM-19, pp. 835–848, October 1971.
- [4] G. C. Clark and J. B. Cain, *Error-Correction Coding for Digital Communications*, New York: Plenum Press, 1981.
- [5] T. Ishitani, K. Tansho, N. Miyahara, and S. Kato, "A Scarce-State-Transition Viterbi-Decoder VLSI for Bit Error Correction," *IEEE Journal of Solid-State Circuits*, vol. SC-22, pp. 575–581, August 1987.

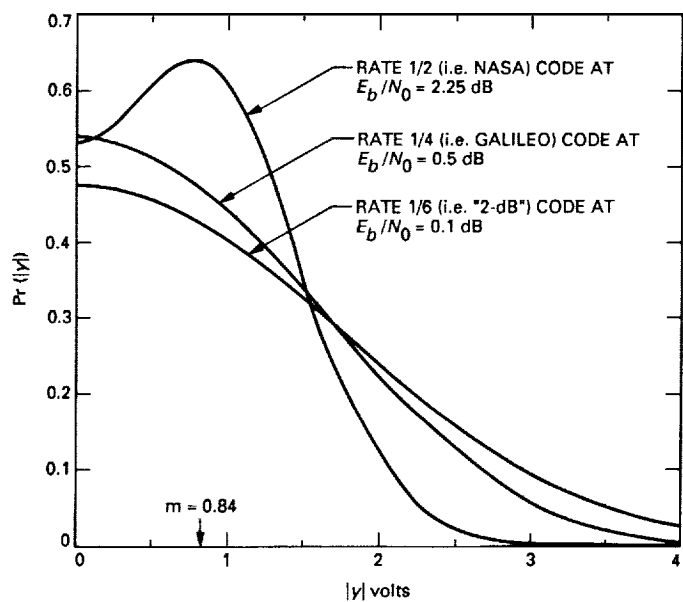


Fig. 1. Received signal magnitude distribution.

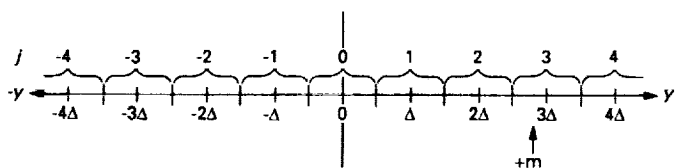


Fig. 2. Optimal quantization for 3-bit branch metrics.

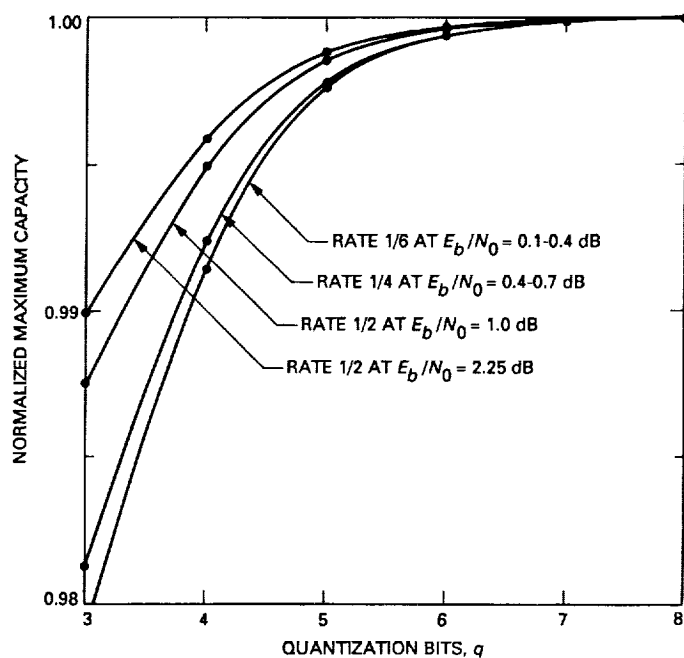


Fig. 3. Uniformly quantized AWGN channel capacities.

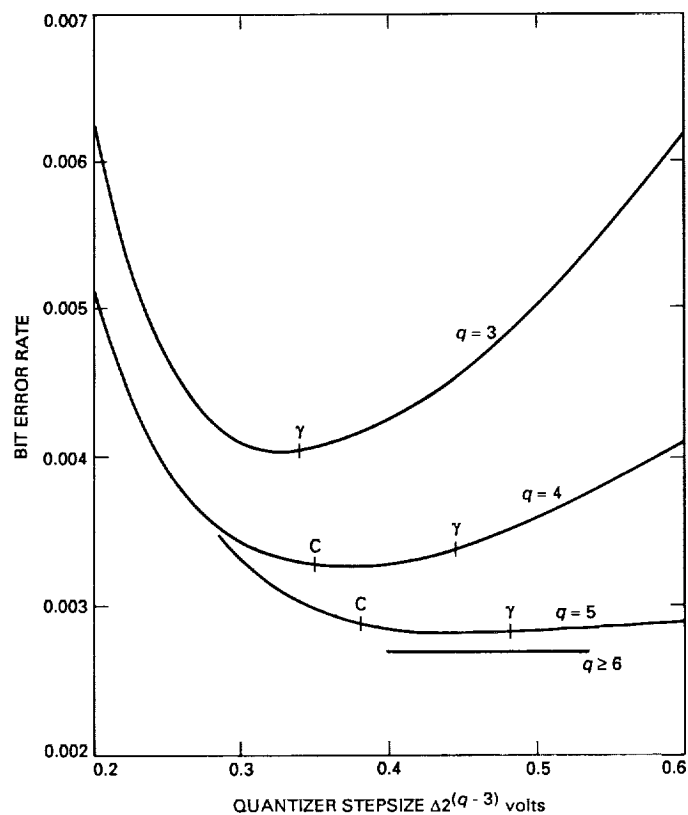


Fig. 4. Performance of the NASA code on the uniformly quantized AWGN channel.

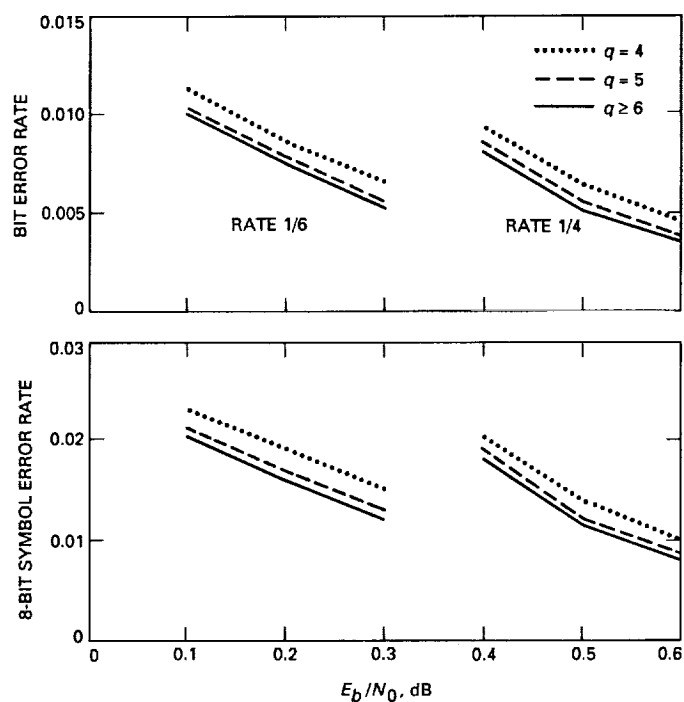


Fig. 5. $K=15$ code simulations.